Non-Conventional Language Models

Instructor: Kyunghyun Cho (NYU, Facebook)

Let's talk BERT[*]

- Three ways to talk BERT
    1. How well does it initialize a classifier?
    2. What kind of (linguistic) features does BERT capture?
    3. What was BERT trained to do?

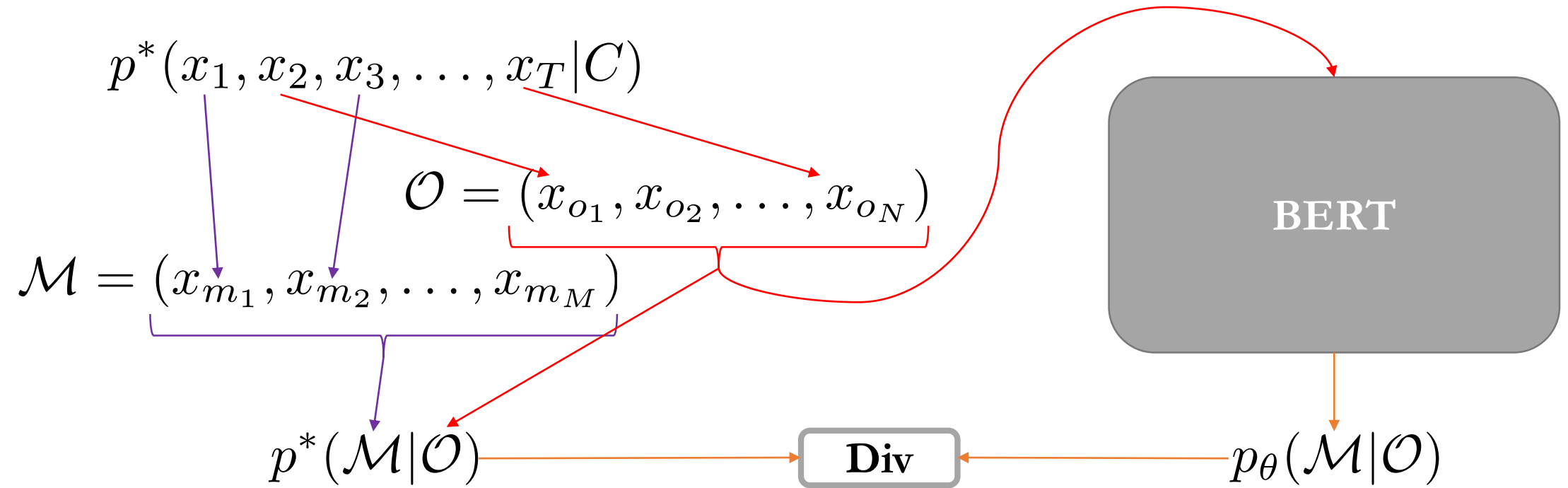* There are many more and less recent variants of BERT, and I use "BERT" to refer to any undirected language model.

How well does it initialize a classifier? – Very

Rank	Name	Model	URL	Score
1	SuperGLUE Human Baselines	SuperGLUE Human Baselines	↗	89.8
2	T5 Team - Google	T5	↗	88.9
3	Facebook AI	RoBERTa	↗	84.6
4	IBM Research AI	BERT-mtl		73.5
5	SuperGLUE Baselines	BERT++	BERT++ 5	
		BERT	↗	69.0

Rank	Name	Model
1	ALBERT-Team Google Language	ALBERT (Ensemble)
2	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)
3	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)
4	Facebook AI	RoBERTa
5	XLNet Team	XLNet-Large (ensemble)
6	Microsoft D365 AI & MSR AI	MT-DNN-ensemble
7	GLUE Human Baselines	GLUE Human Baselines
8	Stanford Hazy Research	Snorkel MeTaL
9	XLM Systems	XLM (English only)
10	Zhuosheng Zhang	SemBERT

What was BERT trained to do?

- **BERT learns a distribution** by learning as many conditionals as possible

$$p^*(x_1, x_2, x_3, \ldots, x_T | C)$$

$$\mathcal{O} = (x_{o_1}, x_{o_2}, \ldots, x_{o_N})$$

$$\mathcal{M} = (x_{m_1}, x_{m_2}, \ldots, x_{m_M})$$

BERT

$$p^*(\mathcal{M} | \mathcal{O})$$

Div

$$p_\theta(\mathcal{M} | \mathcal{O})$$

If BERT learns a distribution, can we generate from it?

- Searching for a **directed**, **acyclic graph*** that covers all the variables
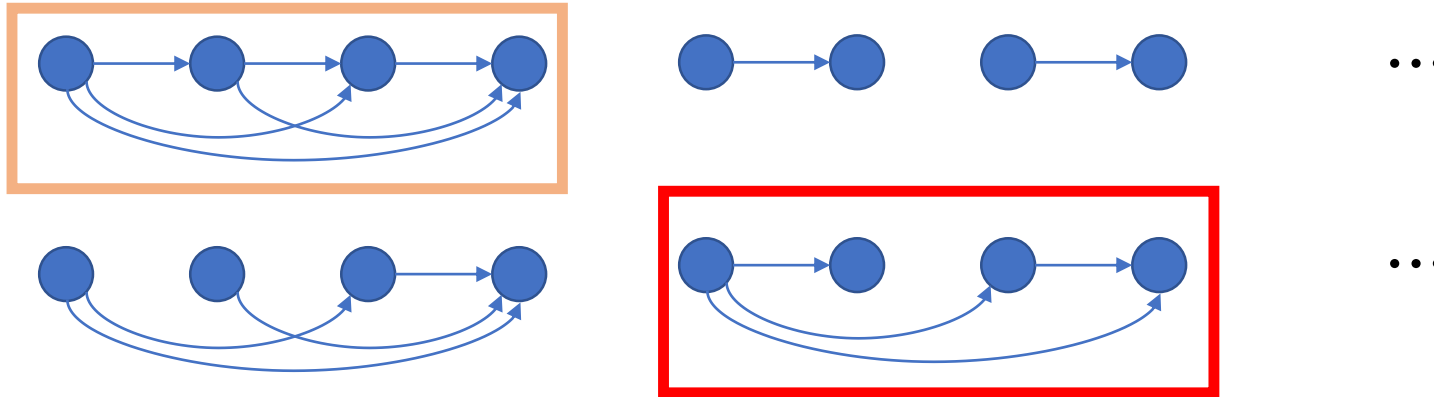
- The optimal **dependency structure** is conditioned on the input C
- Pointed out recently by McAllester as well in his blog post

* it's not necessary to be acyclic in general as long as we
constrain the number of visits per node in topological sweep.

Generalized framework of sequence generation

- Alternates between
    1. Determine the next set of variables to replace
    2. Sample each selected variable
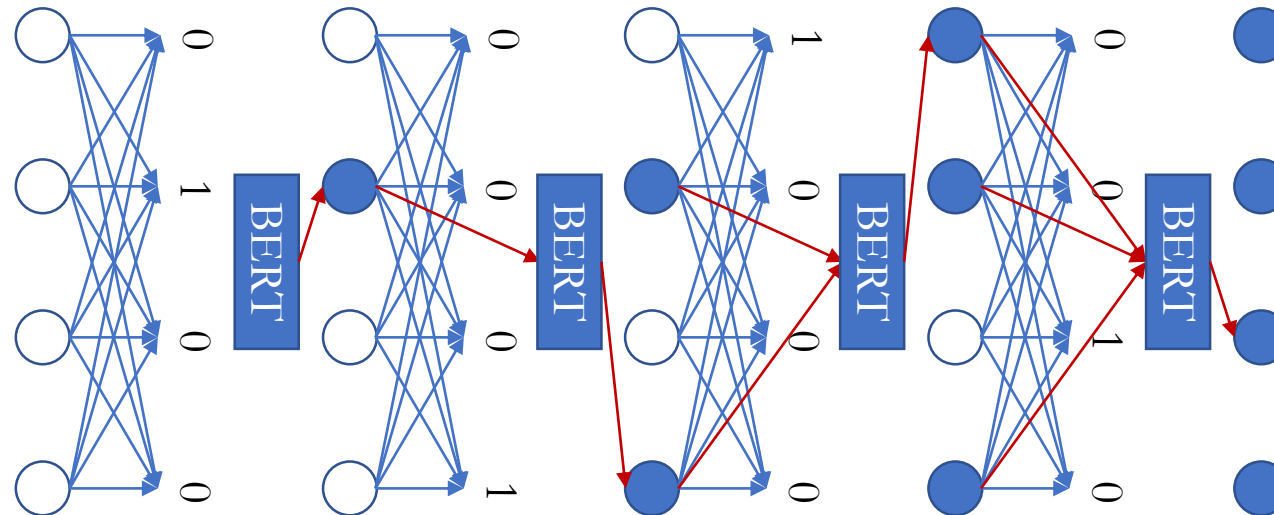
Generalized framework of sequence generation

- Ordering + generation can be framed as

$$\max_{X^{\leq L}, Z^{\leq L}} \left(\sum_{t=1}^{T} \log p(z_t^l | Z^{<l}, X^{<l}, C) \right) + \left(\sum_{t=1}^{T} \log p(x_k | X^{<l}, Z^{<l}, C) \right)$$

where

$$p(z_t^l = 1 | Z^{<l}, X^{<l}, C) = \begin{cases} 0 & \text{if } \sum_{l'=1}^{l-1} z_t^{l'} = 1 \\ f_z^t(Z^{<l}, X^{<l}, C) & \text{otherwise} \end{cases}$$

and

$$p(x_k^l = v | Z^{<l}, X^{<l}, C) = \begin{cases} f_{x,v}^t(Z^{<l}, X^{<l}, C) & \text{if } z_t^l = 0 \\ 1 & \text{otherwise} \end{cases}$$

Mansimov, Wang, Welleck & Cho [2020; under review]

Generation from BERT = Dependency* Detection

$$\max_{X^{\leq L}, Z^{\leq L}} \left(\sum_{t=1}^{T} \log p(z_t^l | Z^{<l}, X^{<l}, C) \right) + \left(\sum_{t=1}^{T} \log p(x_k | X^{<l}, Z^{<l}, C) \right)$$

- BERT has already learned conditionals

- How do we decide a set of variables at each iteration?

Mansimov, Wang, Welleck & Cho [2020; under review]

Confession of a deep learner

- A log-linear model with manually crafted features and coefficients

$$p(z_t^l = 1 | Z^{<l}, X^{<l}, C) \propto \exp \left\{ \sum_{k=1}^{K} \alpha_k \log \phi_k(t, l, \text{BERT}(x^l | X^{<l})) \right\}$$

- Features
  1. Monotonicity: $\log \phi_{\text{loc}}(t, l, \text{BERT}(x^l | X^{<l})) = l - t$
  2. Predictability: $\log \phi_{\text{pred}}(t, l, \text{BERT}(x^l | X^{<l})) = \log p_{\text{BERT}}(x_t^l = x_t^{l-1} | X^{<l})$
  3. Uncertainty: $\log \phi_{\text{ent}}(t, l, \text{BERT}(x^l | X^{<l})) = \log \mathcal{H}_{\text{BERT}}(x_t^l | X^{<l})$

- Coefficients
  - Manually set them to induce desired properties
  - Uniform, left-to-right, easy-first, least-to-most, …

Mansimov, Wang, Welleck & Cho [2020; under review]

Beam search & adaptive generation order

- Because of our formulation, we can beam search with BERT
  - Over the choice of words as well as the choice of variables*
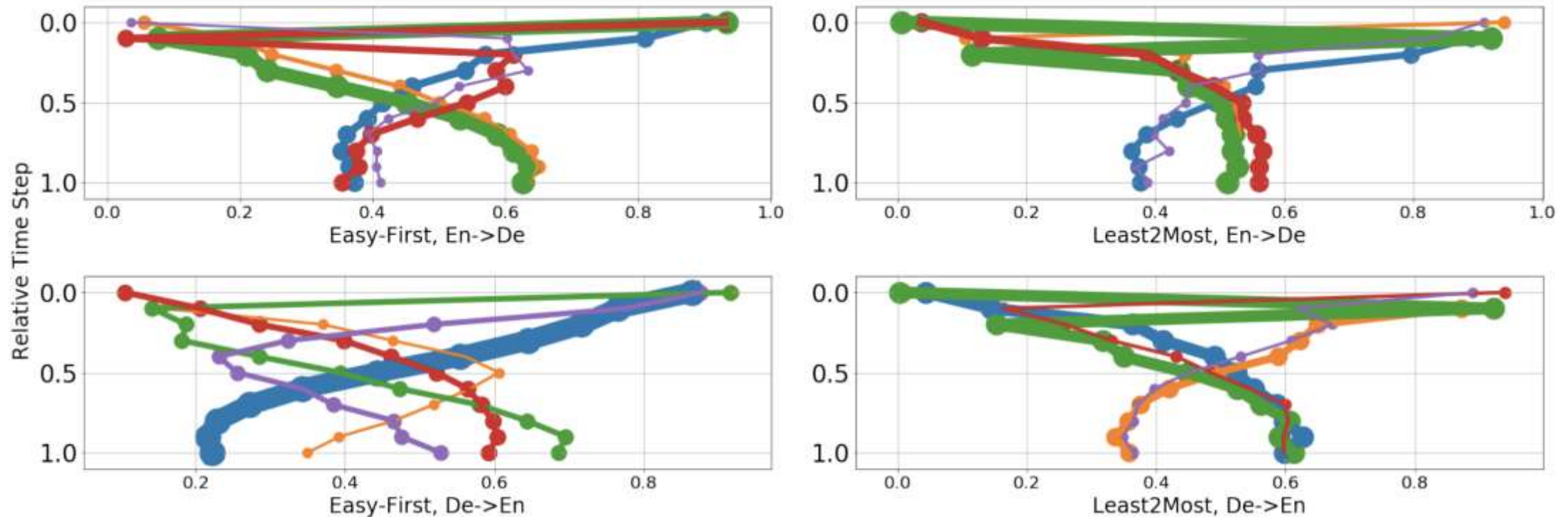
* though, for experiments, we only beam searched over the choice of words.

Source	Doch ohne zivilgesellschaftliche Organisationen könne eine Demokratie nicht funktionieren .
	_ _ _ _ _ _ _ _ _ _ _
	_ _ _ _ _ _ _ _ _ _ .
	_ _ _ _ _ _ _ **cannot** _ .
	_ _ _ _ _ _ _ **democracy** cannot _ .
	_ **without** _ _ _ _ _ democracy cannot _ .
	_ without _ _ _ _ _ democracy cannot **work** .
	But without _ _ _ _ _ democracy cannot work .
	But without _ **society** _ _ _ democracy cannot work .
	But without civil _ _ **,** _ democracy cannot work .
	But without civil **society** _ , _ democracy cannot work .
	But without civil society **organisations** , _ democracy cannot work .
	But without civil society organisations , **a** democracy cannot work .
Reference	Yet without civil society organisations , a democracy cannot function .

Some results

	b	T	Baseline Autoregressive	Decoding from an undirected sequence model				
				Uniform	Left2Right	Least2Most	Easy-First	Learned
En→De	1	L	25.33	21.01	24.27	23.08	23.73	24.10
	4	L	26.84	22.16	25.15	23.81	24.13	24.87
	4	$L*$	–	22.74	**25.66**	24.42	24.69	**25.28**
	1	$2L$	–	21.16	24.45	23.32	23.87	24.15
	4	$2L$	–	21.99	25.14	23.81	24.14	24.86
De→En	1	L	29.83	26.01	28.34	28.85	29.00	28.47
	4	L	30.92	27.07	29.52	29.03	29.41	29.73
	4	$L*$	–	28.07	**30.46**	29.84	30.32	**30.58**
	1	$2L$	–	26.24	28.64	28.60	29.12	28.45
	4	$2L$	–	26.98	29.50	29.02	29.41	29.71

Mansimov, Wang, Welleck & Cho [2020; under review]

But, some interesting implications:
Constant-time machine translation

- With enough parallel compute, each iteration is constant w.r.t. length

$$\hat{Z}^l = \arg\,\mathrm{top} - \lceil L/T \rceil \sum_{t=1}^{T} \log p(\hat{x}_k | \hat{X}^{<l}, \hat{Z}^{<l}, C)$$

$$\hat{X}^l = \arg\max_{X^l} \log p(\hat{z}_t^l | \hat{Z}^{<l}, \hat{X}^{<l}, C)$$

L	K	Uniform	Left2Right	Least2Most	Easy-First	Hard-First
10	$T \to 1$	22.38	22.38	27.14	22.21	26.66
10	$T \to 1^*$	23.64	23.64	28.63	23.79	28.46
10	$\lceil T/L \rceil$	22.43	21.92	24.69	25.16	23.46
20	$T \to 1$	26.01	26.01	28.54	22.24	28.32
20	$T \to 1^*$	27.28	27.28	30.13	24.55	29.82
20	$\lceil T/L \rceil$	24.69	25.94	27.01	27.49	25.56

Initially put forward by Ghazvininejad et al. [2019]

Sequence generation with an adaptive order

- BERT Generation
  - Ghazvininejad et al. and Lawrence et al. at EMNLP 2019
- Learned-order generation
  - Welleck et al. [2019], Gu et al. [2019], Stern et al. [2019], Chan et al. [2019], Emelianenko et al. [2019], etc.

Probabilistic sequence modeling

- $p^*(x_1, x_2, \ldots, x_T | C)$, where $x_t \in V, |V| < \infty, T < \infty$
- From a finite set of data points $D = \{(X_1, C_1), \ldots, (X_N, C_N)\}$
- Three ways to go
  1. ~~Build a table containing all possible X for each C to store probabilities~~
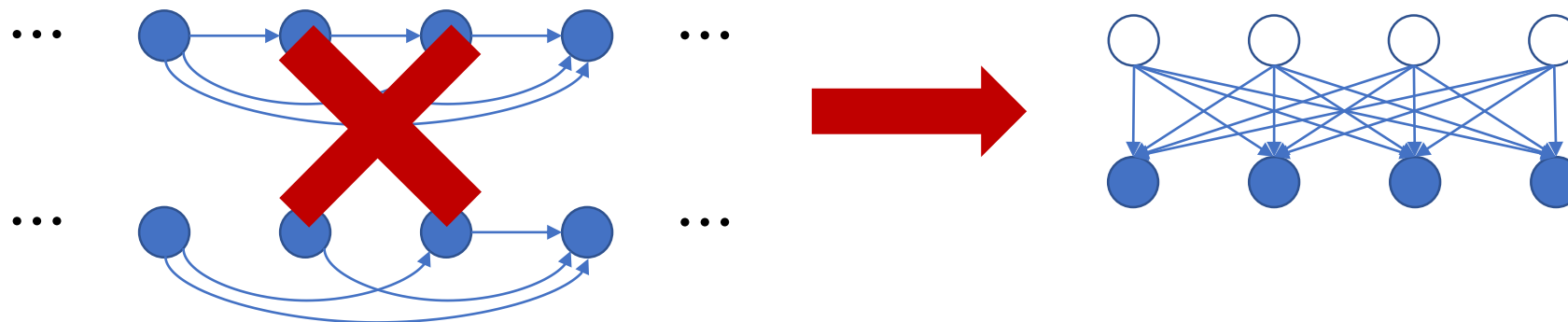 2. Factorize the table: e.g., autoregressive models

$$p(x_1, \ldots, x_T | C) = \prod_{t=1}^{T} p(x_t | x_1, \ldots, x_{t-1}, C)$$

  3. Smooth the table: e.g., latent variable models

$$p(x_1, \ldots, x_T | C) = \int_{\mathcal{Z}} p(x_1, \ldots, x_T | Z, C) p(Z | C) \mathrm{d}Z$$

Latent variable models

- Latent variables capture the dependencies

- $x_i \perp\!\!\!\perp x_j | z_{1:T}$, but after marginalization $x_i \not\!\perp\!\!\!\perp x_j$

Latent variable models

- Notoriously difficult to fit a latent variable model
  - Principal component analysis [Pearson, 1901], Boltzmann machines [Ackley et al., 1985], Helmholtz free energy [Hinton, 1994], wake-sleep algorithm [Hinton et al., 1995], independent component analysis [Comon, 1994; Bell&Sejnowski, 1995; Hyvarinen et al., 2000], contrastive divergence [Hinton, 2002], deep belief nets [Hinton et al., 2006], denoising autoencoders [Vincent et al., 2009], … and my entire PhD dissertation…

- Because, it's difficult to marginalize the latent variables

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log \int_{z} p(x^n | z, C^n) p(z, C^n) \mathrm{d}z$$

Variational autoencoders

- Maximizing the lowerbound instead with an approximate posterior q

$$\log \int_z p(x|z,C)p(z|C)\mathrm{d}z \geq \mathbb{E}_{z \sim q(z|x,C)}\left[\log p(x|z,C)\right] - \mathrm{KL}(q(z|x)\|p(z|C))$$

  - Reparametrization trick to estimate the gradient w.r.t. q with a minimal variance
  - Proposed by Kingma & Welling [2013] and Rezende et al. [2014]

- A stream of studies applying VAE to NLP
  - VAE Language models [Bowman et al., 2016],
    VAE Seq2seq [Zhang et al., 2016; Zhou&Neubig, 2017; Shah&Barber, 2018], …

Learning is easy, but generation is not…

- Generation is equivalent to solving
$$\hat{x} = \arg \max_x \mathbb{E}_{z \sim q(z|x,C)} \left[\log p(x|z, C) \right] - \mathrm{KL}(q(z|x) \| p(z|C))$$

- It is *not* equivalent to $\arg \max_x \log p(x | \mathbb{E}_q[z], C)$

- Sadly, this has been a common practice
  - *"During decoding, however, due to the absence of target sentence y, we set h_z to be the mean of $p_\theta(z|x)$, i.e., μ'."* Zhang et al. [2016]
  - *"We do not marginalize over the latent variable z however, instead we use the mode μ of z as the latent representation for z"* Zhou&Neubig [2017]

Deterministic inference algorithm

- Delta posterior

$$q'(z|x,C) = \begin{cases} 1, & \text{if } z = \mu \\ 0, & \text{otherwise} \end{cases}$$
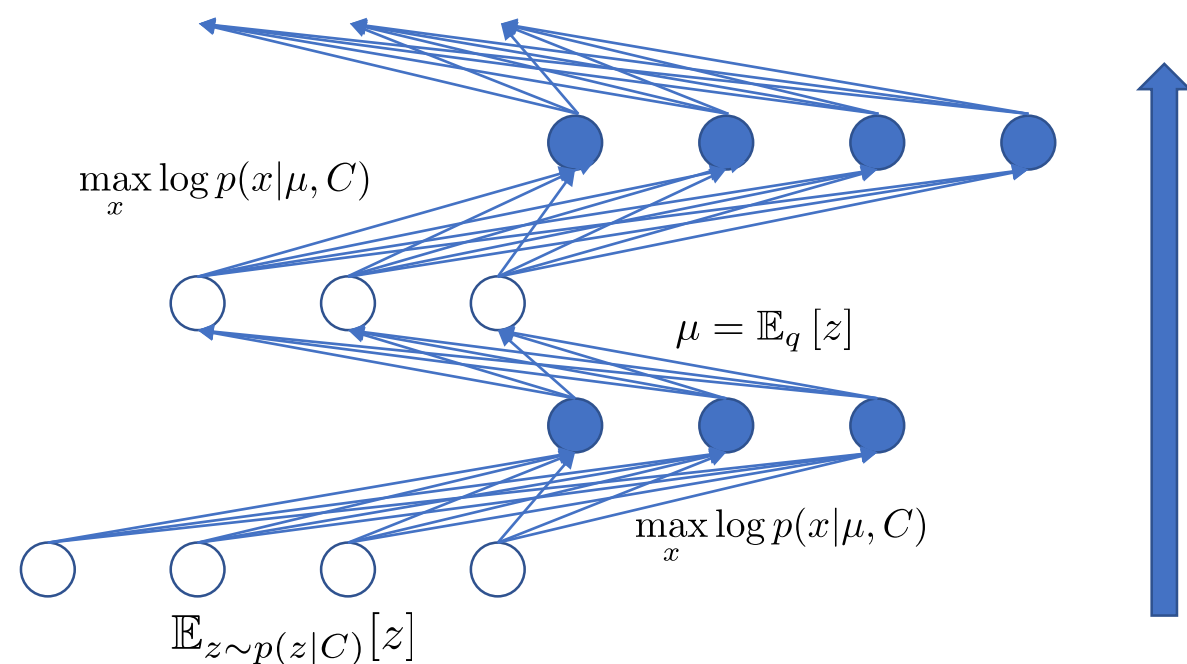
- Alternating between two steps
                    until convergence

  1. Fit the delta posterior

  $$\min_{\mu} \text{KL}(q'\|q) \iff \mu = \mathbb{E}_q[z]$$
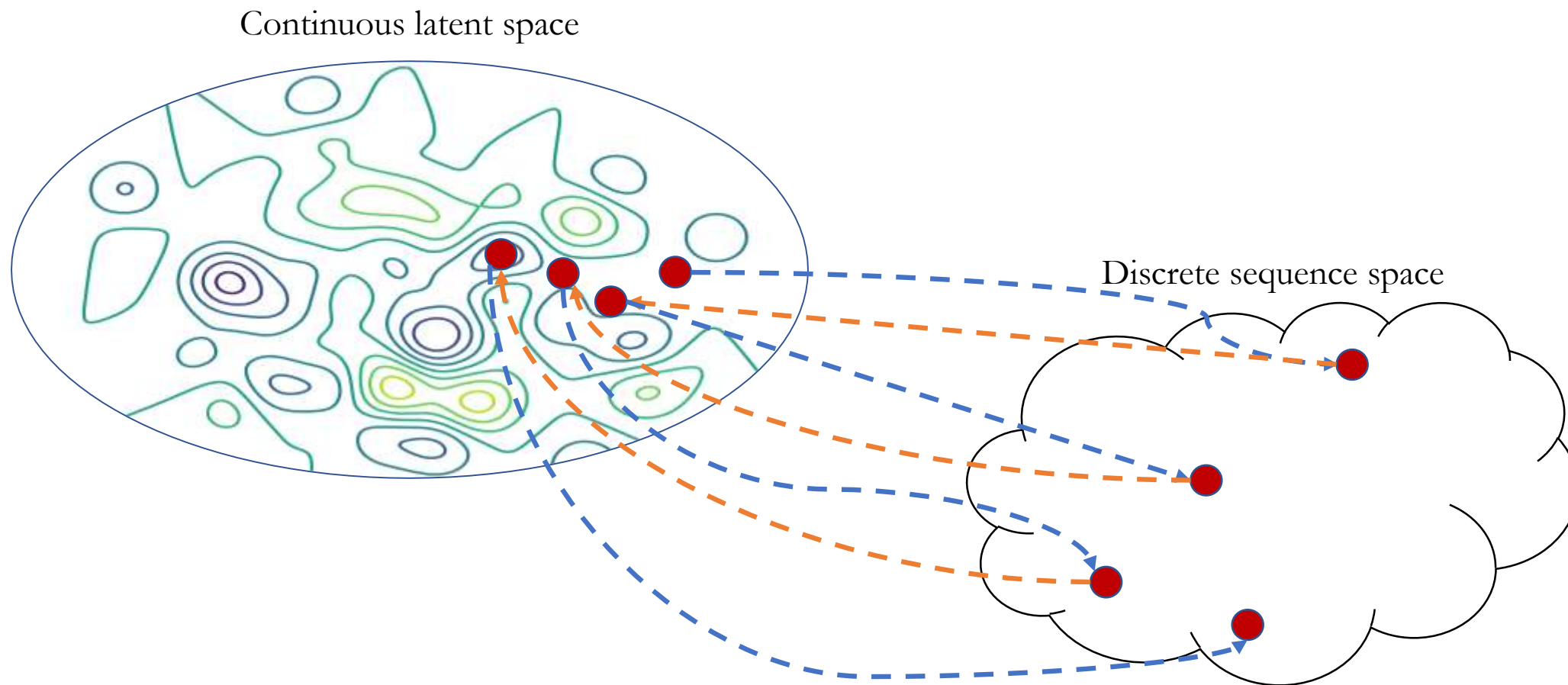
  2. Maximize the lowerbound using q'

  $$\max_{x} \mathbb{E}_{q'}[\log p(x|z,C)] = \max_{x} \log p(x|\mu,C)$$

$$\max_{x} \log p(x|\mu,C)$$

$$\mu = \mathbb{E}_q[z]$$

$$\max_{x} \log p(x|\mu,C)$$

$$\mathbb{E}_{z\sim p(z|C)}[z]$$

[Shu et al., 2019; rejected]

A stochastic version of this algorithm was proposed by Shah&Barber [2018]. So, **novelty** 👎

Alternating optimization enables big jumps

Continuous latent space

Discrete sequence space

Latent variables allow inference to make jumps

## Example 1: Sequence modified without changing length

Source	hyouki gensuiryou hyoujun no kakuritsu wo kokoromita. (Japanese)
Reference	the establishment of an optical fiber attenuation standard was atte
Initial Guess	an attempt was made establish establish damping attenuation standar
After Inference	an attempt was to establish the damping attenuation standard ...

Example 2: One word removed from the sequence

|	
Source	...``sen bouchou keisu no toriatsukai'' nitsuite nobeta. (Japanese)
Reference	... handling of linear expansion coefficient .
Initial Guess	... `` handling of of linear expansion coefficient '' are described
After Inference	... `` handling of linear expansion coefficient '' are described .

Example 3: Four words added to the sequence

Source	... maikuro manipyureshon heto hatten shite kite ori ...(Japanese)
Reference	... with wide application fields so that it has been developed ...
Initial Guess	... micro micro manipulation and ...
After Inference	... and micro manipulation , and it has been developed , and ...
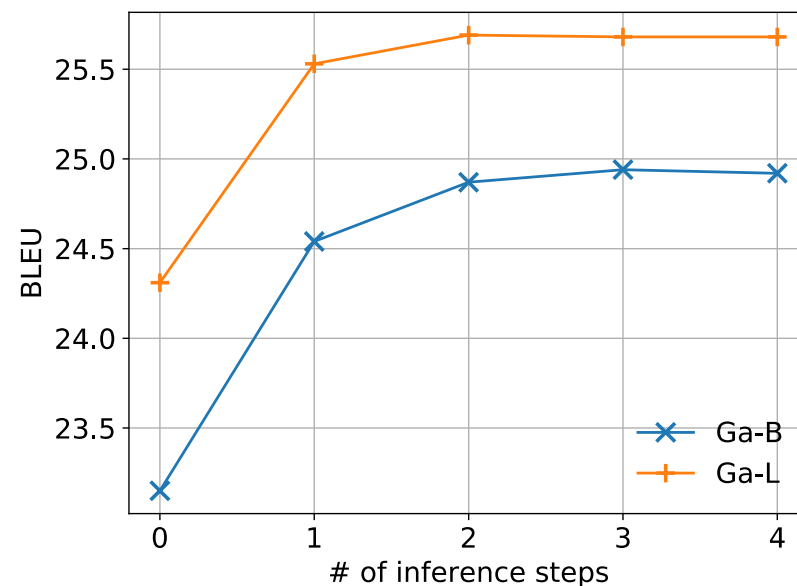
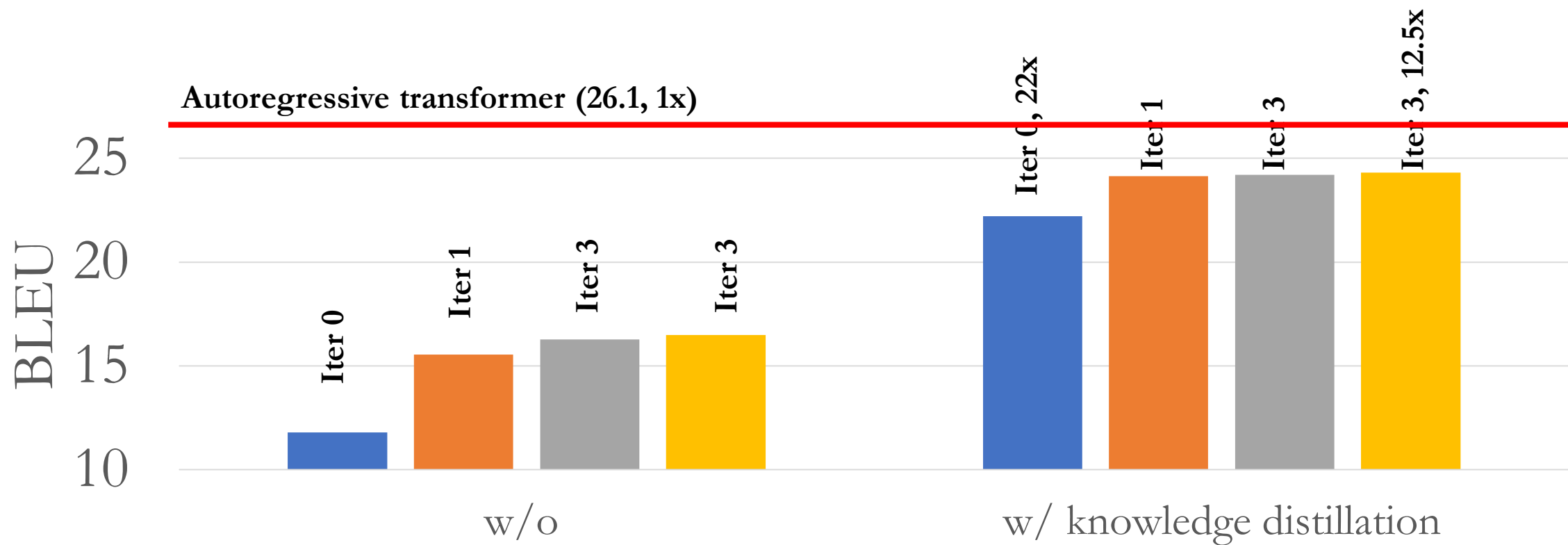

[Shu et al., 2020; Lee et al., 2020]
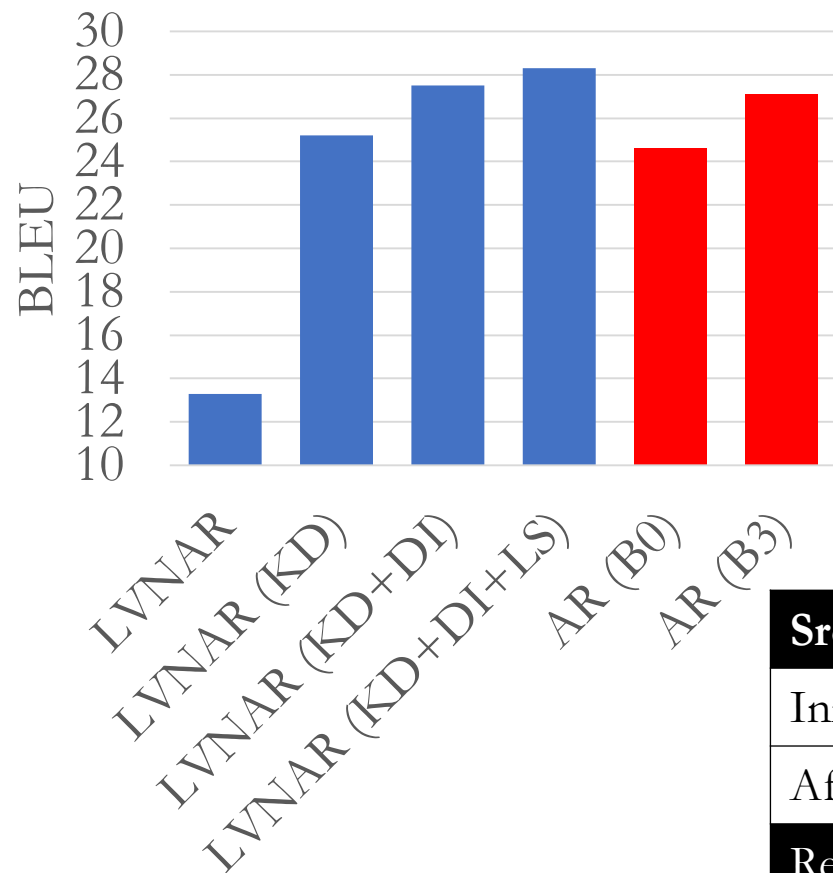
Better quality-efficiency trade-off
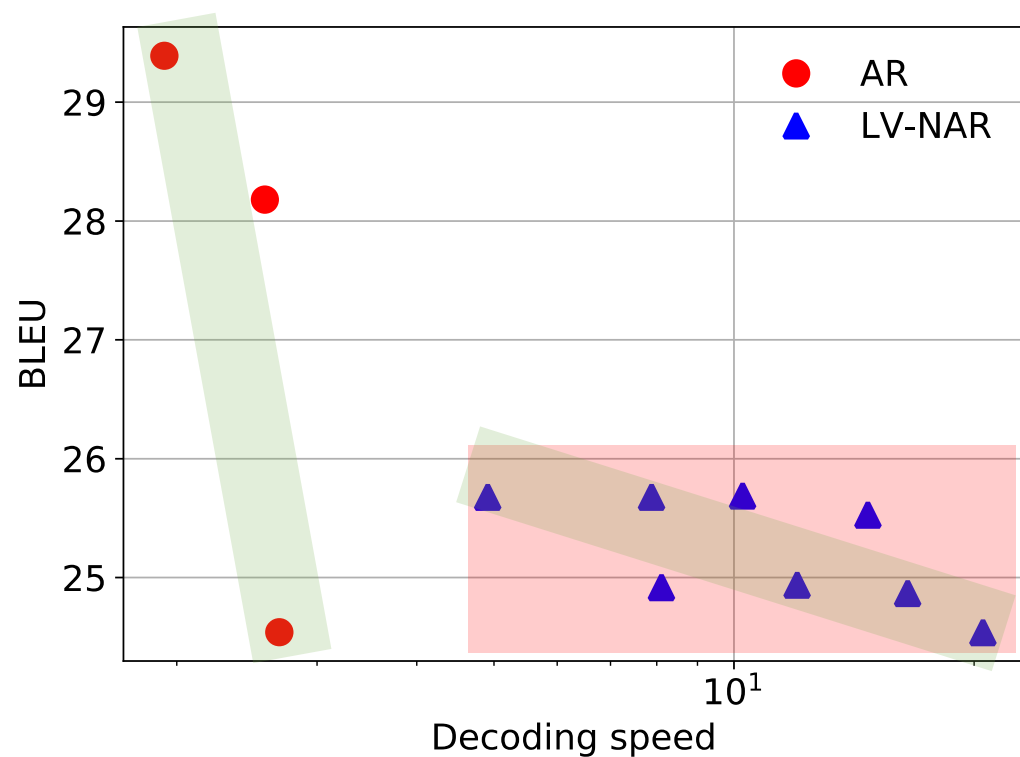
[Lee et al., 2020]

- Continuous space results in a *better* tradeoff between efficiency and quality

- Continuous space provides us with a more *fine-grained control* over the trade-off

Still **not state-of-the-art**, but pretty efficient

Autoregressive transformer (26.1, 1x)

BLEU

w/o

w/ knowledge distillation

Iter 0
Iter 1
Iter 3
Iter 3

Iter 0, 22x
Iter 1
Iter 3
Iter 3, 12.5x

[Shu et al., 2020]

Almost **state-of-the-art** in a simpler domain

- Ja-En ASPEC [Nakazawa et al., 2016]
  - Limited to scientific domains
  - A narrower set of sentence styles

Src	標記 減 衰 量 標準 の 確立 を 試み た 。
Initial	an attempt was to establish the damping attenuation standard …
After	an attempt **tries** to establish the damping **quantity** standard …
Ref	an attempt was made establish establish damping attenuation standard …

In this lecture, we learned

- Undirected (masked) language models
  - How to generate from a masked language model
  - How to learn the order of generation from a masked language model

- Latent-variable non-autoregressive machine translation
  - How to construct a latent-variable model
  - How to perform generation using iterative refinement